

SOA技術の最新動向

— より柔軟にビジネス・ニーズに応えるために —

本稿では SOA における基幹技術の代表である Enterprise Service Bus (以下、ESB) と Business Process Execution Language (以下、BPEL) の概要とビジネスの柔軟性をどのようにサポートしてきたかについて述べるとともに、ビジネス・ニーズにより柔軟に応えるための技術として最近注目されているイベント処理、ビジネス・ルールとポリシーに関する概要、さらにはそれらの使われ方についてご紹介いたします。

① 今日の SOA 技術

今日、SOA を支える代表的な技術は何かと問われると、多くの方は ESB と BPEL を挙げられると思います。それぞれの技術に関する詳細については割愛し、本章では多くのお客様でどのような使われ方をしているかについてまとめたいと思います。

1.1 ESB

SOA というと、誰もが ESB という技術を思い浮かべるかと思えます。ESB の目的はアプリケーション間連携の柔軟性を提供することにあります。アプリケーションが持つビジネス的な機能をサービスという概念で公開し、そこにオープン・スタンダード技術 (Web サービス) を適用することで、サービス・プロバイダー (サービスの提供者) 側が提供しているサービスの実装 (サービス・インターフェースや ESB との接続プロトコル) をサービス・コンシューマー (サービスの利用者) から隠すことにあります (図 1)。これによりサービス・コンシューマーが、サービス・プロバイダーの提供するサービスを直接呼ばずに済むため、サービス・プロバイダー側の実装に依存しない実装が可能になります。このサービス・インターフェースのマップを行うのが ESB 内にあるメディエーションという機能です。このメディエーションにより、単なるデータ項目のマップのみならず、データの Join/Split のような加工、データ型の変換、データ内のマスター・コードの変換、サービス・プロバイダーの動的選択、複数サービス・プロバイダー

The Latest Trends in SOA Technology - To Respond to Business Needs More Flexibly -

This article gives an overview of Enterprise Service Bus (hereafter, ESB) and Business Process Execution Language (hereafter, BPEL), major SOA technologies, and describes how they have been contributing to business flexibility. As a future technology trend, this article covers an overview of event processes and business rules & policies, and their business values.

への配信などを連携させることができます。

この ESB の使い方に関する大きな誤解は、サービス・プロバイダーが提供するサービス・インターフェースは Web サービスだけというものです。実は ESB はさまざまな形態のサービス・プロバイダーが持つインターフェース (バッチ・アプリケーション、ホスト・アプリケーション、ERP などのパッケージ・アプリケーション、データ・ベース、FTP、e-メール) をサービスとしてサービス・コンシューマーに公開する機能を持っています。よって、SOA に基づいてサービス・コンシューマーを開発する際、Web サービスを提供するサービス・プロバイダーは必ずしも必要ではありません。多くのお客様の場合、ESB とアダプターの組み合わせにより既存のアプリケーションをサービス化することで、効率よく SOA を適用したソリューションを構築されています。このように既存のアプリケーションを生かしながらバッチ・アプリケーションと連携する場合に ESB のパフォーマンスを心配する方が数多くいます。そのような使い方に関しても、100 点ではありませんが、昨今の ESB 製品は大量データにも耐えられるスケーラビリティを持った製品が出てきています。例えばある製造のお客様で

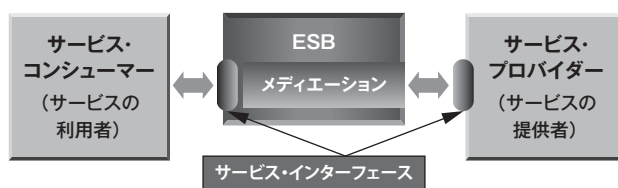


図1. ESBの概要

は、1日当たりのピークが120GBを超えるメッセージやファイルのB2BとERP間連携を、ESBを通して行っています。また、このように既存のさまざまなインターフェースをサービス化することにより、段階的なレガシー・システムのSOA化も可能になります。IBM自身もこのようなアプローチを社内のシステムに適用して、ビジネスのニーズに応じて段階的な社内システムのSOA化に取り組んでいます。

1.2 BPEL

SOAの柔軟性では、アプリケーション間連携に次いでビジネス・プロセスの柔軟性が求められます。多くのお客様においては、さまざまなアプリケーションと手作業の組み合わせでビジネス・プロセスが構成されています。これらのアプリケーションが提供する業務的な機能と手作業のフローを制御する機能としてBPELがあります。BPELはWebサービス呼び出しのフローを制御するXMLで定義された言語で、図2に示すようにほかのアプリケーションと手作業に関するWebサービス呼び出しのフロー制御を行います。BPELは現在OASISで制定されたオープン・スタンダードの技術です。この技術と人間系の作業を管理するコンポーネント(図2のヒューマン・タスク・マネージャー)を組み合わせることで、ワーク・フローとアプリケーションとの連携フローを一つの言語で表記できるようになりました。

さて、このような機能は10年以上も前よりワーク・フロー

技術として存在していましたが、当時はまだXML、WebサービスそしてBPELなどのオープン・スタンダードがなく、それぞれのベンダーが独自の技術で実装していました。特に他アプリケーションとの連携の方法はOSごとに異なるため、他アプリケーションとの接続性は悪く、プロセス・フローを柔軟に変更することはかなりコストがかかる作業でした。ESBを通したプロセス・フローとアプリケーションの連携により、接続先の変更はかなり容易になり、ソリューションの実装や更改にかかるコストは劇的に減りました(図2)。例えば、あるお客様では40ステップにわたるESBを通した他システムとの連携も含めたプロセス・フローのソリューションを、ほぼ5カ月で開発しました。このようにBPELは他システムとの連携や人間系のワーク・フローを組み合わせたソリューションを短期間の内に開発することを可能にします。

2 最新の動向

これまでのSOA技術の流れは、今まで個々のアプリケーションの中に含まれていた、アプリケーション間連携や業務プロセス・フローをオープン・スタンダードに準拠したミドルウェアの上に乗せ、アプリケーション・ロジックから切り離すことを一つの目標にしてきました。この基本的な方向性は今後も続いていきますが、その中でSOA技術は以下のような進化を遂げています。

- イベント処理
 - ビジネス・ルールとポリシー
- 以降、この2つの技術動向と使われ方について紹介していきます。

2.1 イベント処理

従来のSOAソリューションの基本は、ESBを中心にしたメッセージ・ベースのアプリケーション間連携にあります。ここでいうメッセージには宛先が付いていますが、イベントには宛先がありません。その代わりに、イベントの中には「いつ」「どこで」「どのような状態で」「何が起きた」といった情報が入ります。そして複数のイベント・ソースから発信されたイベントは、イベント処理

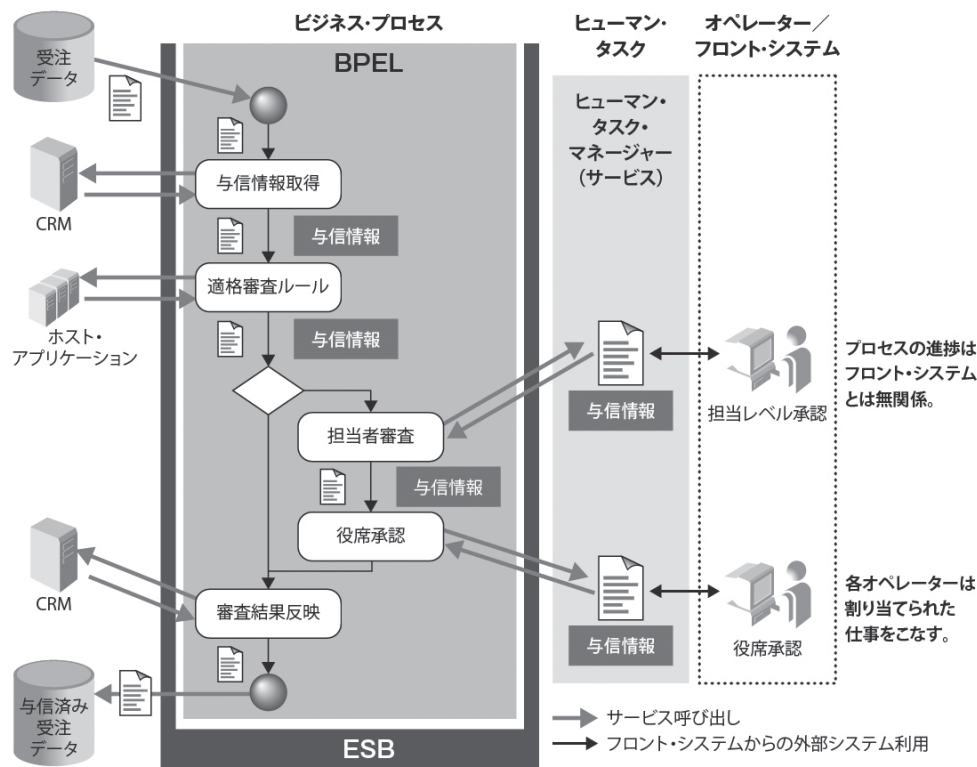


図2. BPELによるビジネス・プロセスの連携

ネットワーク内でのイベント処理を経て、最終的には複数のイベント・コンシューマーによって処理（アクションの実行）されます（図3）。

イベント処理の典型的な例としては、2カ所で使われたクレジットカードのケースが挙げられます。同じクレジットカードが博多と札幌で1時間以内に使われたとしましょう。この場合博多と札幌の店という2つのイベント・ソースからそれぞれ1つのイベントが発せられます。各イベントには使われたクレジットカード番号、購入金額、店名、時間などの情報が入っています。この2つのイベントはイベント処理ネットワーク内において、以下のような処理が行われます。

1. 店名から判別される位置情報をイベントに追加
2. クレジット・カード番号で関連付けを行い、それぞれのイベントにある購入時間と位置情報からクレジットカードの不正使用（どちらかもしくは両方）の可能性を検知
3. クレジット・カード不正使用アラート・イベントの発行

このクレジットカード不正使用アラート・イベントを監視しているイベント・コンシューマーが、このアラート・イベントを受けて何らかのアクションを取る（Web サービスを呼ぶ）こととなります。ここで取られるアクションの一つにはビジネス・プロセスがあります。この例の場合、アラート・イベントを受け取ったイベント・コンシューマーは、eメールもしくはコール・センターを通してクレジットカードの所有者にコンタクトし、それぞれのクレジットカードの使用が正しいかどうかを確認するなどのビジネス・プロセスを Web サービス呼び出しにより自動的に起動します。以上のように、SOA にイベント処理を追加することにより、ビジネスの現場で起きているさまざまな状況とその状況に対処するためのビジネス・プロセスを連動させることが可能になります。半導体の製造工程の中ではすでにこのようなソリューションが稼働していて、製造ラインの停止時間の削減などのビジネス的な効果を出しています。

2.2 ビジネス・ルールとポリシー

まずルールとポリシーの違いについて、例を使って簡単に紹介します。ポリシーの例としては、「ゴールド顧客になるためには年間100万円の購入が必要」というものが挙げられます。これに対してルールの例は、「ゴールド顧客が月10万円以上購入したら、次月の購入に対して15%の割引が適用される」というものです。このようにポリシーは制

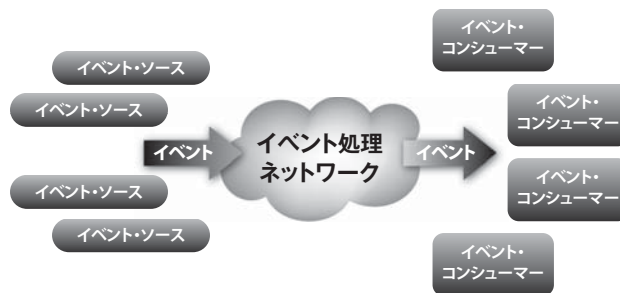


図3. イベント処理

約条件もしくは満たすべき条件であり、ルールはある入力に対して出力を生成するための一連の条件と結果（上記の例では「15%の割引が適用」）となります。以降、SOA におけるビジネス・ルールとポリシーについて紹介していきます。

① ビジネス・ルール

従来のアプリケーションでは、先に例として挙げたようなルールをアプリケーション・コードの一部として組み込む方法や、ある決まった形式のルールとしてマスター化する方法によって、最小限の柔軟性を実装してきました。このようなアプローチは、昨今のビジネスのニーズに対して機敏に応えることができなくなってきています。例えば、競争に勝ち抜くために複雑化する携帯電話やインターネットの料金プラン、あるいは保険契約などにおいては、対象としているデータ構造やルールの構造の変更が必要な場合が多々見受けられます。また、ビジネス・ルールの編集そのものについても、ビジネス・ユーザーが直接変更して運用に回すといった機能が必要になってきました。そこで出てきたのが、ビジネス・ルール・エンジンです。図4に示すように、アプリケーションの中に組み込まれていたビジネス・ルールを外出しにして、ルール変更に伴うアプリケーションの書き換えを防ぎ、ビジネ

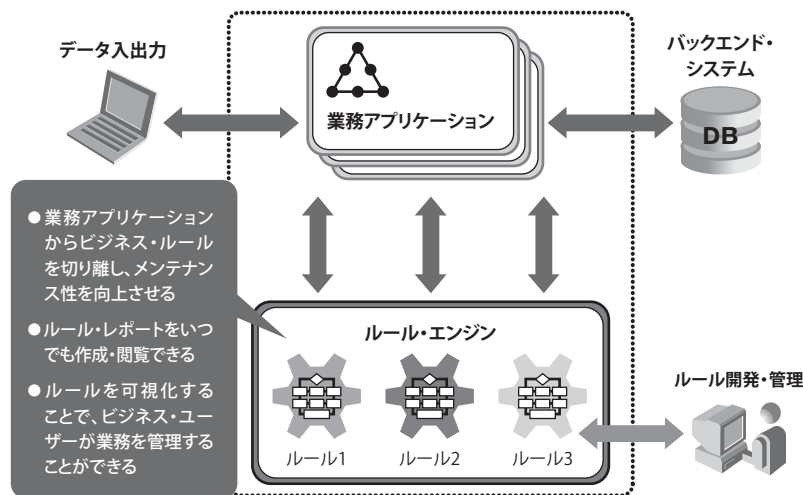


図4. ビジネス・ルール・エンジン

ス・ユーザーにそれらのルールの管理・編集を任せることができるようになります。

② ポリシー

SOA におけるポリシーの適用に関しては、以下に示すように大きく分けて 2 つあります。

- 個々のサービスを管理するためのサービス・ポリシー
- ビジネス・プロセスを抽象化するためのビジネス・ポリシー

サービス・ポリシー

サービス・ポリシーの定義は、サービスの再利用の促進やライフサイクル管理のために、業務、アーキテクチャーそして運用管理の領域でそれぞれに行われます。以下にこれらの領域におけるポリシーの例を示します。

業務領域ポリシー

- 個人を特定可能な情報は保護されること（セキュリティー & プライバシー・ポリシー）
- 金融取引が内部統制に準拠していること（業務ポリシー）

アーキテクチャー領域ポリシー

- 管理責任者は担当する顧客の信用レポートにのみアクセス可能（アプリケーション・セキュリティー・ポリシー）
- サービスは独自のものであることが保障されている（開発ライフサイクル・ポリシー）

運用管理領域ポリシー

- メッセージと承認処理に対するデジタル署名によるセキュリティー（アクセス管理ポリシー）

- それぞれのメッセージは確実に 1 度だけ届けられる（メッセージングの信頼性ポリシー）

以上のような 3 つの領域のポリシーにより、既存サービス・アセットを再利用する際に、その使用の目的に合致しているかについて確認することができます。

ビジネス・ポリシー

ビジネス・プロセスを実装する BPEL 技術が世の中に出てきて 10 年近くたちますが、すでに多くのお客様において BPEL を使った数多くのソリューションが作られてきました。その中で分かってきたことの一つに、さまざまな要因から条件分岐が増加し、BPEL で実装されたプロセスが複雑化（入れ子も増える）していき、ということが挙げられます。つまり取り扱われる商品やその商品が取り扱われる地域によって、対処の仕方がプロセスの中で変わるといったことによりプロセスが複雑化しているのです。当然複雑化したもの（入れ子になった一連の条件文）の変更およびテストは困難を極め、開発工数や期間も膨らみます。

こうした問題を解決するために、ビジネス・ポリシーが活用されます。ビジネス・プロセスを上位ビジネス・プロセス、サブ・プロセス、そしてサービス・エンドポイントの 3 層に分け、それぞれ下位層を呼び出す際にビジネス・ポリシーを使うことで、上位層のプロセスに下位層を呼び分ける条件文を組み込むことなく、複雑なビジネス・プロセスを単純化することが可能になります（図 5）。

このようなビジネス・プロセスの中で使われるポリシーはリポジトリで管理され、プロセスを止めることなく動的に変更

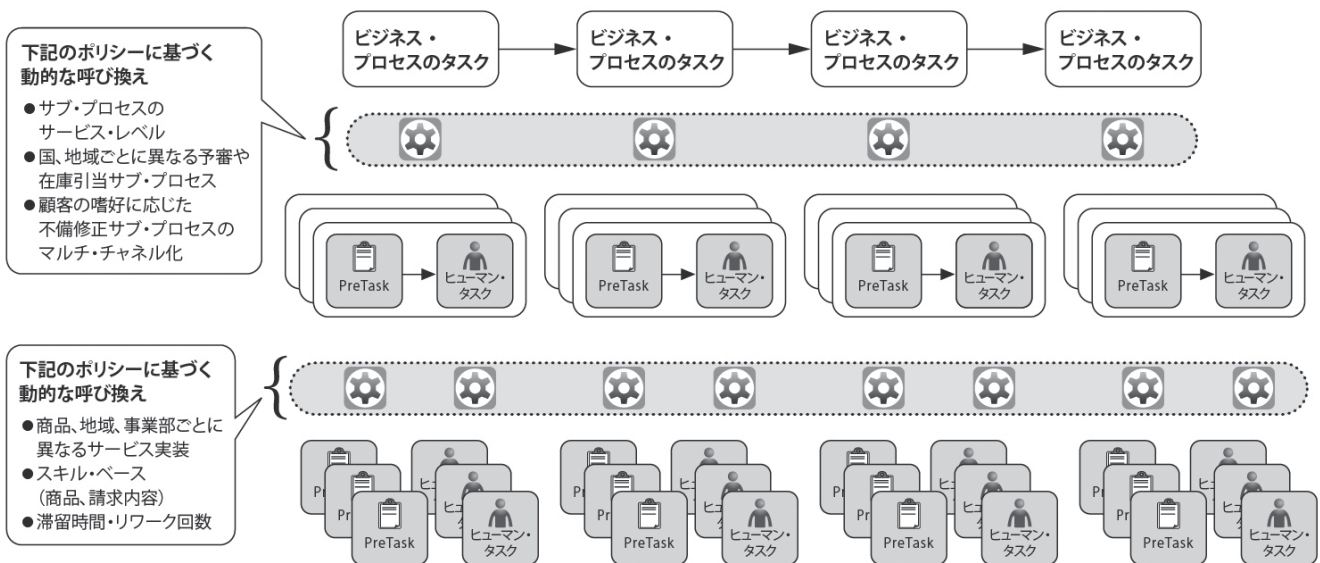


図5. ビジネス・ポリシーの活用によるBPM

することができるため、ビジネス・ニーズに応じて柔軟に変更することができるようになっていきます。

③ 最後に

今回紹介いたしました SOA/BPM の技術ですが、すでに数多くのお客様で使用されているものです。では、いかにこのような新しい技術を実際の業務に適用され、そのビジネス的な効果を楽しんでいるのでしょうか。多くのお客様で共通にいえることは、“Think Big, Start Small” ということです。すなわち、適用技術の選定やソリューションのアーキテクチャーに関しては将来を見据えて大胆に考え決めていく。しかし最初のプロジェクトは 3～6 カ月でソリューションを構築し、短期間でビジネス的なリターンを確実に得て、次のプロジェクトにつなげていく。そしてこうした実践を通して横展開することにより、大きな目標に近づいていくというものです。このときの大きなポイントは、既存のアプリケーションを最大限活用するというものです。実際に日本のお客様の成功事例でも同様のアプローチにより、その効果を実感され横展開を図っています。

本稿が皆さまの SOA/BPM の適用の一助になることを祈り締めくらせていただきます。



日本アイ・ビー・エム株式会社
ソフトウェア事業
ソフトウェア・テクニカルセールス&サービス
シニア・テクニカル・スタッフ・メンバー、
SOA/BPM アーキテクト

吉田 洋一 Yoichi Yoshida

【プロフィール】

1984 年日本 IBM に入社。入社以来 3270 データ・ストリーム・アーキテクチャーや System Application Architecture の日本における推進と日本語サポートに従事。SOA/BPM に関しては 2001 年より大和ソフトウェア開発研究所において関連製品開発のアーキテクトとして従事。2004 年アメリカ駐在中に Business Activity Monitor 製品である WebSphere Business Monitor の設計にアーキテクトとして従事。2006 年より、SOA/BPM アーキテクトとして日本のお客様における SOA/BPM の適用の支援に従事している。